

CLAIMS

What is claimed is:

1. A method for providing an automatic initial load of data to a new table copy concurrently both with user changes to a source table copy to the new table copy and with active replication to existing table copies in an asynchronous replication group, comprising:

(a) initiating a data load of the new table copy at a source table copy;

5 (b) creating a spill storage area at the new table copy;

(c) loading the data from the source table copy to the new table copy, wherein changes for the active table copies in the replication group are applied to the active table copies during the loading, wherein changes to the new table copy are stored in spill storage area during the loading;

10 (d) applying the changes in the spill storage area to the new table copy after the loading of the data is done; and

(e) removing the spill storage area after the changes in the spill storage area have been applied to the new table copy.

15 2. The method of claim 1, wherein the initiating (a) comprises:

(a1) seeing a start signal at the source table copy;

(a2) sending a schema message from the source table copy to the new table copy;

and

(a3) changing a state for a new table copy subscription at the source table copy

20 from inactive to loading.

3. The method of claim 1, wherein the creating (b) comprises:

(b1) receiving a schema message at the new table copy;

(b2) invoking a load utility at the new table copy, if the data load is an internal data load; and

5 (b3) creating the spill storage area.

4. The method of claim 3, wherein the creating (b) further comprises:

(b4) changing a state of a new table copy subscription at the new table copy from “inactive” to “internal loading”, if the data load is an internal data load; and

10 (b5) changing the state of the new table copy subscription at the new table copy from “inactive” to “external loading”, if the data load is an external data load.

5. The method of claim 1, wherein the loading (c) comprises:

(c1) loading the data from the source table copy to the new table copy;

15 (c2) applying the changes for the active table copies to the active table copies during the data load;

(c3) storing the changes for the new table copy in the spill storage area during the data load; and

(c4) sending a load done message from the new table copy to the source table copy
20 after the data load is done, if the data load is an internal load.

6. The method of claim 5, wherein the loading (c) further comprises:

(c5) receiving the load done message at the source table copy;

(c6) sending the load done message back to the new table copy; and
(c7) changing a state of a new table copy subscription at the source table copy
from “loading” to “active”.

5 7. The method of claim 6, wherein the loading (c) further comprises:

(c8) tagging the changes for the new table copy that occur before the load done
message is received at the source table copy.

8. The method of claim 1, wherein the applying (d) comprises:

10 (d1) receiving a load done message at the new table copy;

(d2) starting a spill agent thread at the new table copy;

(d3) changing a state of a new table copy subscription at the new table copy to
“load finishing”; and

(d4) processing the changes stored in the at least one spill queue by the spill agent
15 thread.

9. The method of claim 8, wherein for an internal data load, a thread used to
load the data is reused as the spill agent thread.

20 10. The method of claim 8, wherein the processing (d4) comprises:

(d4i) re-executing the changes to a same row at the new table copy in a same order
as at the source table copy;

(d4ii) identifying and resolving dependencies between the changes to the same row;

and

(d4iii) detecting and resolving conflicts between the changes to the same row.

11. The method of claim 10, wherein the re-executing (d4i) comprises:

5 (d4iA) deleting the re-executed change from the spill storage area in a same transaction as the re-execution.

12. The method of claim 11, wherein the deleting (d4iA) is performed with a two-phase commit protocol.

10 13. The method of claim 11, wherein the deleting (d4iA) comprises:

(d4iA(I)) storing a message identifier for the re-executed change at the new table copy, wherein upon a restart of the data load at the new table copy, only changes from the spill storage area without a corresponding stored message identifier are applied to the new
15 table copy.

14. The method of claim 10, wherein the detecting and resolving (d4iii) comprises:

(d4iiiA) ignoring a change to a row not found in the new table copy when re-
20 executing a conflicting row delete;

(d4iiiB) ignoring a change to a row not found in the new table copy when re-executing a conflicting row update;

(d4iiiC) ignoring a change to a row in the new table copy when re-executing a

conflicting row insert; and

(d4iiiD) ignoring a change to an old row not found or a new row found in the new table copy when re-executing a conflicting key update.

5 15. The method of claim 10, wherein the detecting and resolving (d4iii) comprises:

(d4iiiA) ignoring a missing row in the new table copy when re-executing a row delete;

10 (d4iiiB) transforming a re-execution of an update into a row insert when a row in the new table copy is missing; and

(d4iiiC) transforming a re-execution of an insert into an update when a row in the new table copy already exists.

16. The method of claim 1, wherein the removing (e) comprises:

15 (e1) sending a spill queue empty message by a spill agent thread at the new table copy when the spill agent thread reaches a last change on the spill storage area;

(e2) determining by a browser thread at the new table copy that an oldest running transaction at the new table copy is older than a most recent transaction when a load done message was received at the source table copy;

20 (e3) sending a spill final message from the browser thread to the spill agent thread; and

(e4) deleting the spill storage area when emptied by the spill agent thread.

17. The method of claim 16, wherein the removing (e) further comprises:

(e5) sending a spill done message by the spill agent thread to the browser thread;

(e6) terminating the spill agent thread;

(e7) receiving the spill done message by the browser thread; and

5 (e8) changing a state of a new table copy subscription at the new table copy to
“active”.

18. A computer readable medium with program instructions for providing an
automatic initial load of data to a new table copy concurrently both with user changes to a
10 source table copy to the new table copy and with active replication to existing table copies in
an asynchronous replication group, comprising:

(a) initiating a data load of the new table copy at a source table copy;

(b) creating a spill storage area at the new table copy;

(c) loading the data from the source table copy to the new table copy, wherein
15 changes for the active table copies in the replication group are applied to the active table
copies during the loading, wherein changes to the new table copy are stored in the spill
storage area during the loading;

(d) applying the changes in the spill storage area to the new table copy after the
loading of the data is done; and

20 (e) removing the spill storage area after the changes in the spill storage area have
been applied to the new table copy.

19. The medium of claim 18, wherein the initiating (a) comprises:

(a1) seeing a start signal at the source table copy;
(a2) sending a schema message from the source table copy to the new table copy;
and
(a3) changing a state for a new table copy subscription at the source table copy
5 from inactive to loading.

20. The medium of claim 18, wherein the creating (b) comprises:
(b1) receiving a schema message at the new table copy;
(b2) invoking a load utility at the new table copy, if the data load is an internal
10 data load; and
(b3) creating the spill storage area.

21. The medium of claim 20, wherein the creating (b) further comprises:
(b4) changing a state of a new table copy subscription at the new table copy from
15 “inactive” to “internal loading”, if the data load is an internal data load; and
(b5) changing the state of the new table copy subscription at the new table copy
from “inactive” to “external loading”, if the data load is an external data load.

22. The medium of claim 18, wherein the loading (c) comprises:
20 (c1) loading the data from the source table copy to the new table copy;
(c2) applying the changes for the active table copies to the active table copies
during the data load;
(c3) storing the changes for the new table copy in the spill storage area during the

data load; and

(c4) sending a load done message from the new table copy to the source table copy after the data load is done, if the data load is an internal load.

5 23. The medium of claim 22, wherein the loading (c) further comprises:

(c5) receiving the load done message at the source table copy;

(c6) sending the load done message back to the new table copy; and

(c7) changing a state of a new table copy subscription at the source table copy from “loading” to “active”.

10

24. The medium of claim 23, wherein the loading (c) further comprises:

(c8) tagging the changes for the new table copy that occurs before the load done message is received at the source table copy.

15 25. The medium of claim 18, wherein the applying (d) comprises:

(d1) receiving a load done message at the new table copy;

(d2) starting a spill agent thread at the new table copy;

(d3) changing a state of a new table copy subscription at the new table copy to “load finishing”; and

20 (d4) processing the changes stored in the at least one spill queue by the spill agent thread.

26. The medium of claim 25, wherein for an internal data load, a thread used to

load the data is reused as the spill agent thread.

27. The medium of claim 25, wherein the processing (d4) comprises:

(d4i) re-executing the changes to a same row at the new table copy in a same order
5 as at the source table copy;

(d4ii) identifying and resolving dependencies between the changes to the same row;
and

(d4iii) detecting and resolving conflicts between the changes to the same row.

10 28. The medium of claim 27, wherein the re-executing (d4i) comprises:

(d4iA) deleting the re-executed change from the spill storage area in a same
transaction as the re-execution.

28. The medium of claim 28, wherein the deleting (d4iA) is performed with a

15 two-phase commit protocol.

30. The medium of claim 28, wherein the deleting (d4iA) comprises:

(d4iA(I)) storing a message identifier for the re-executed change at the new table
copy, wherein upon a restart of the data load at the new table copy, only changes from the
20 spill storage area without a corresponding stored message identifier are applied to the new
table copy.

31. The medium of claim 27, wherein the detecting and resolving (d4iii)

comprises:

(d4iiiA) ignoring a change to a row not found in the new table copy when re-executing a conflicting row delete;

(d4iiiB) ignoring a change to a row not found in the new table copy when re-executing a conflicting row update;

(d4iiiC) ignoring a change to a row in the new table copy when re-executing a conflicting row insert; and

(d4iiiD) ignoring a change to an old row not found or a new row found in the new table copy when re-executing a conflicting key update.

32. The medium of claim 27, wherein the detecting and resolving (d4iii)

comprises:(d4iiiA) ignoring a missing row in the new table copy when re-executing a row delete;

(d4iiiB) transforming a re-execution of an update into a row insert when a row in the new table copy is missing; and

(d4iiiC) transforming a re-execution of an insert into an update when a row in the new table copy already exists.

33. The medium of claim 18, wherein the removing (e) comprises:

(e1) sending a spill queue empty message by a spill agent thread at the new table copy when the spill agent thread reaches a last change on the spill storage area;

(e2) determining by a browser thread at the new table copy that an oldest running transaction at the new table copy is older than a most recent transaction when a load done

message was received at the source table copy;

(e3) sending a spill final message from the browser thread to the spill agent thread; and

(e4) deleting the storage area by the spill agent thread.

5

34. The medium of claim 33, wherein the removing (e) further comprises:

(e5) sending a spill done message by the spill agent thread to the browser thread;

(e6) terminating the spill agent thread;

(e7) receiving the spill done message by the browser thread; and

10 (e8) changing a state of a new table copy subscription at the new table copy to “active”.

35. A system, comprising:

at least one active table copy in an asynchronous replication group, wherein the at

15 least one active table copy comprises a source table copy;

a new table copy; and

a spill storage area at the new table copy,

wherein a data load to the new table copy is initiated at the source table copy,

wherein the spill storage area is created after the data load is initiated,

20 wherein the data is loaded from the source table copy to the new table copy,

wherein changes for the at least one active table copy are applied to the at least one active table copy during the loading of the data,

wherein changes for the new table copy are stored in the spill storage area during the

loading of the data,

wherein the changes in the spill storage area are applied to the new table copy after the loading of the data is done, and

wherein the spill storage area is removed after the changes in the spill storage area have been applied to the new table copy.

5